AMENDMENTS TO THE CLAIMS

1. (Previously presented) A method of automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, the method comprising the computer-implemented steps of:

receiving a location value that identifies a location of the keystream segment within the complete keystream;

creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node.

2. (Previously presented) A method as recited in Claim 1, further comprising the steps of creating and storing the balanced binary tree by creating and storing a stack of h elements wherein the i^{th} element of said stack stores a state datum for the i^{th} node on a path from a root node of the tree to the leaf node.

3. (Previously presented) A method as recited in Claim 2, wherein the step of creating and storing a state value for a leaf node comprises the steps of computing and storing a state value for the leaf node that is unique with respect to any other state value that is computed at any other time for any other leaf node of the tree.

4. (Canceled.)

5. (Previously presented) The method as recited in Claim 1, wherein each leaf node stores m bits of state information, wherein m is a multiple of twelve.

6. (Currently amended) The method as recited in Claim 1, further comprising the steps of:

creating and storing $m=3n$ bits of state information in each leaf node comprising a concatenation of three n bit quantities $z | y | x$, wherein n is a multiple of four;

computing the first non-linear function a and the second non-linear function b as the composition of a diffusion function d with the nonlinear "confusion" functions f and g, wherein $a = f \circ d$ and $b = g \circ d$ and wherein

$$f(z \mid y \mid x) = 2z \mid S(R(S(R(y)))) \mid L(S(L(S(x))))$$

$$g(z \mid y \mid x) = 2z+1 \mid L(S(L(S(y)))) \mid S(R(S(R(\neg x))))$$

$$d(z \mid y \mid x) = z \mid x + y + z \mid 2x + y + z$$

$$c(z \mid y \mid x) = x \oplus y$$

wherein integer addition modulo two is denoted as +, bitwise exclusive-or is denoted as $\oplus$, and bitwise complementation is denoted as $\neg$;

wherein the R denotes rotation by $n/4$ bits to in a direction of a least significant bit and L denotes rotation by $n/4$ bits in a direction of a most significant bit; and

wherein a nonlinear function S comprises a lookup in a key-dependent substitution table.

7. (Previously presented) The method as recited in Claim 1, wherein the third function comprises computing a linear reduction of $2n$ bits of the state value to n bits thereof.

8. (Previously presented) A method as recited in Claim 6, wherein the third function comprises computing a bitwise Boolean exclusive OR of x and y.

9. (Previously presented) A method as recited in Claim 6, further comprising the steps of creating and storing the substitution table *S* by selecting four invertible functions and storing the four invertible functions in a concatenated form.

10. (Previously presented) A method as recited in Claim 6, further comprising the steps of computing functions *f* and *g* in seven instructions of a central processing unit that can issue two instructions simultaneously, by using five registers to store values of *x*, *y*, *z*, a temporary variable, and a pointer to the substitution table *S*.

11. (Previously presented) A method as recited in Claim 6, wherein the substitution table *S* comprises an array of key dependent pseudorandom integer values.

12. (Previously presented) A method as recited in Claim 6, wherein the substitution table *S* comprises an array of 256 key dependent pseudorandom 32-bit unsigned integer values.

13. (Previously presented) The method as recited in Claim 1, further comprising the steps of creating and storing a key for use by the first non-linear function and the second non-linear function, wherein the key comprises a table of key dependent pseudorandom values.

14. (Previously presented) The method as recited in Claim 1, further comprising the steps of creating and storing, once and at a time prior to receiving the location value, a key for use by the first non-linear function and the second non-linear function, wherein the key comprises a table of key dependent pseudorandom values.

15. (Previously presented) The method as recited in Claim 1, further comprising the steps of creating and storing a key in the form of a plurality of pseudo-randomly selected invertible functions, wherein each of the invertible functions maps an 8-bit portion of the state value to an 8-bit quantity for use as a substitute portion of the state value.

16. (Previously presented) A method as recited in Claim 1,wherein the pseudo-randomly selected invertible functions are stored in a plurality of substitution tables, and wherein the plurality of substitution tables are generated by:
setting each of the plurality of substitution tables equal to the identity function;
for each element of each of the plurality of substitution tables, swapping said element with another element of such table in a key-dependent manner, and also performing the same swapping operation on each table that has been previously been generated.

17. (Previously presented) A method of enciphering a plaintext using at least one keystream segment at an arbitrary location of a complete keystream, the method comprising the computer-implemented steps of:
receiving a segment of a plaintext;
receiving a location value that identifies a location of the keystream segment within the complete keystream;
creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node;

enciphering the segment of the plaintext by combining the keystream segment with the segment of the plaintext using a Boolean exclusive OR operation to result in creating and storing a segment of ciphertext.

18. (Previously presented) A method of encrypting an ordered plurality of packets of a network communication link using at least one keystream segment at an arbitrary location of a complete keystream, the method comprising the computer-implemented steps of:

receiving a packet from among the plurality of packets;

determining a location value that represents a relative location of the packet among the plurality of packets;

creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents a keystream segment at the relative location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node;

enciphering the packet by combining the keystream segment with data of the packet using a Boolean exclusive OR operation to result in creating and storing enciphered packet data.

19. (Previously presented) A computer-readable medium carrying one or more sequences of instructions for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, which

instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving a location value that identifies a location of the keystream segment within the complete keystream;

creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node.

20. (Previously presented) An apparatus for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, comprising:

means for receiving a location value that identifies a location of the keystream segment within the complete keystream;

means for creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

means for creating and storing the keystream segment by applying a third function to the state value of the leaf node.

21. (Amended) An apparatus for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, comprising:

a network interface that is coupled to the data network for receiving one or more packet flows therefrom;

a processor;

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

receiving a location value that identifies a location of the keystream segment within the complete keystream;

creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node.

22. (New) A computer-readable medium as recited in Claim 19, comprising further sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of any of Claims 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, or 16.

23. (New) An apparatus as recited in Claim 20, further means for performing functions recited in the steps of any of Claims 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, or 16.

24. (New) An apparatus as recited in Claim 21, wherein the sequences of instructions comprise further sequences of instructions which, when executed by the processor, cause the processor to perform the steps of any of Claims 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, or 16.